Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 395 (2008) 268–282
Theoretical Computer Science
www.elsevier.com/locate/tcs

Computing similarity of run-length encoded strings with affine gap penalty☆

Jin Wook Kim[a], Amihood Amir[b,c], Gad M. Landau[d,e], Kunsoo Park[a,*]

[a] *School of Computer Science and Engineering, Seoul National University, Seoul, 151-742, Republic of Korea*
[b] *Department of Computer Science, Bar-Ilan University, 52900 Ramat-Gan, Israel*
[c] *Johns Hopkins University, United States*
[d] *Department of Computer Science, University of Haifa, Mount Carmel, Haifa 31905, Israel*
[e] *Polytechnic University, United States*

Abstract

The problem of computing the similarity of two run-length encoded strings has been studied for various scoring metrics. Many algorithms have been developed for the longest common subsequence metric and some algorithms for the Levenshtein distance metric and the weighted edit distance metric. In this paper we consider similarity based on the affine gap penalty metric which is a more general and rather complicated scoring metric than the weighted edit distance. To compute the similarity in this model efficiently, we convert the problem into a path problem on a directed acyclic graph and use some properties of maximum paths in this graph. We present an $O(nm' + n'm)$ time algorithm for computing the similarity of two run-length encoded strings in the affine gap penalty model, where n and m are the lengths of given two strings whose run-length encoded lengths are n' and m', respectively.

Keywords: Run-length encoding; Affine gap penalty; Similarity

1. Introduction

A string S is *run-length encoded* if it is described as an ordered sequence of pairs (σ, i), often denoted "σ^i", each consisting of an alphabet symbol σ and an integer i [2]. Each pair corresponds to a *run* in S, consisting of i consecutive occurrences of σ. Let A and B be two strings with lengths n and m, respectively. Let A' and B' be two run-length encoded strings of A and B, and n' and m' be the lengths of A' and B', respectively.

The problem of computing the similarity of two run-length encoded strings, A' and B', has been studied for various scoring metrics. For the longest common subsequence metric, Bunke and Csirik [3] presented an $O(nm' + n'm)$ time algorithm, while Apostolico, Landau, and Skiena [1] gave an $O(n'm' \log(n'm'))$ time algorithm and Mitchell [13]

☆ K. Park's work was supported by FPR05A2-341 of 21C Frontier Functional Proteomics Project from Korean Ministry of Science & Technology and A. Amir and G. M. Landau's work was partially supported by the Israel Science Foundation grant 35/05.
* Corresponding author. Tel.: +82 2 880 8381; fax: +82 2 885 3141.
E-mail address: kpark@theory.snu.ac.kr (K. Park).

0304-3975/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.01.008

Table 1
Various scoring metrics

Metric	Match	Mismatch	Indel	Indel of k characters
Longest common subsequence	1	0	0	0
Levenshtein distance	0	1	1	k
Weighted edit distance	0	δ	μ	$k\mu$
Affine gap penalty	1	$-\delta$	$-\gamma-\mu$	$\gamma-k\mu$

obtained an $O((d+n'+m')\log(d+n'+m'))$ time algorithm, where d is the number of matches of compressed characters. Mäkinen, Navarro, and Ukkonen [12] conjectured an $O(n'm')$ time algorithm on average without proof.

For the Levenshtein distance metric, Arbell, Landau, and Mitchell [2] and Mäkinen, Navarro, and Ukkonen [11] presented $O(nm'+n'm)$ time algorithms, independently. Mäkinen, Navarro, and Ukkonen [11] posed as an open problem the challenge of extending these results to more general scoring metrics. Crochemore, Landau, and Ziv-Ukelson [5,4] and Mäkinen, Navarro, and Ukkonen [12] gave $O(nm'+n'm)$ time algorithms for the weighted edit distance metric using techniques completely different from each other.

In this paper we consider similarity based on the affine gap penalty metric. Table 1 shows the differences between four metrics. The affine gap penalty metric is the most general of the four and it is a rather complicated scoring metric than the weighted edit distance. To compute the similarity in this model efficiently, we convert the problem into a path problem on a directed acyclic graph and we use some properties of paths in this graph to select only *essential paths*, i.e., the paths that must be considered to compute entries correctly. It is not necessary to build the graph explicitly since we come up with new recurrences from the essential paths.

We present an $O(nm'+n'm)$ time algorithm for computing similarity of two run-length encoded strings in the affine gap penalty model, where n and m are the lengths of given two strings whose run-length encoded lengths are n' and m', respectively. This result shows that we successfully extended comparison of run-length encoded strings to a more general scoring metric.

The remainder of the paper is organized as follows. In Section 2, we describe the global alignment algorithm due to Gotoh, convert the alignment problem into a path problem on a directed acyclic graph, and give some properties of maximum paths in this graph. In Section 3, we present new recurrences for a *white block* and a *black block* using the properties of the graph and give an efficient algorithm for computing similarity of two run-length encoded strings in the affine gap penalty model based on the new recurrences. We conclude in Section 4.

2. Preliminaries

We first give some definitions and notations that will be used in this paper. A string is concatenations of zero or more characters from an alphabet Σ. A space is denoted by $\Delta \notin \Sigma$; we regard Δ as a character for convenience. The length of a string A is denoted by $|A|$. Let a_i denote the ith character of a string A and $A[i..j]$ denote substring $a_i a_{i+1} \dots a_j$ of A. When a string α is a substring of string A, we denote it by $\alpha \prec A$. Given two strings $A = a_1 a_2 \dots a_n$ and $B = b_1 b_2 \dots b_m$, an alignment of A and B is $A^* = a_1^* a_2^* \dots a_l^*$ and $B^* = b_1^* b_2^* \dots b_l^*$ constructed by inserting zero or more Δs into A and B so that each a_i^* maps to b_i^* for $1 \le i \le l$. There are three kinds of mappings in a^* and b^* according to the characters of a_i^* and b_i^*.

- match : $a_i^* = b_i^* \neq \Delta$,
- mismatch : $(a_i^* \neq b_i^*)$ and $(a_i^*, b_i^* \neq \Delta)$,
- insertion or deletion (indel for short) : either a_i^* or b_i^* is Δ.

Note that we do not allow the case of $a_i^* = b_i^* = \Delta$.

2.1. Global alignments

Given two strings A and B, let $SG(A, B)$ denote the *similarity* of A and B, which we define formally below. Informally, it is the similarity score of an optimal global alignment between A and B.

A well-known algorithm to find an optimal global alignment was given by Waterman et al. [14] and Gotoh [7]. Given two strings A and B where $|A| = n$ and $|B| = m$, the algorithm computes $SG(A, B)$ using a dynamic programming table (called the H *table*) of size $(n+1)(m+1)$. Let H_{ij} for $0 \le i \le n$ and $0 \le j \le m$ denote $SG(A[1..i], B[1..j])$. Then, H_{ij} can be computed by the following recurrence:

$$\begin{aligned} H_{i,0} &= -g_i \quad \text{for } 0 \le i \le n \\ H_{0,j} &= -g_j \quad \text{for } 0 \le j \le m \\ H_{ij} &= \max\left\{H_{i-1,j-1} + s(a_i, b_j), C_{ij}, R_{ij}\right\} \quad \text{for } 1 \le i \le n,\ 1 \le j \le m \end{aligned} \tag{1}$$

where g_k, $s(a_i, b_j)$, C_{ij}, and R_{ij} are defined as follows.

- g_k is the gap penalty for an indel of $k \ge 1$ bases such that $g_k = \gamma + k\mu$ where γ and μ are non-negative constants.
- $s(a_i, b_j)$ is the similarity score between elements a_i and b_j such that

$$s(a_i, b_j) = \begin{cases} 1 & \text{if } a_i = b_j \\ -\delta & \text{if } a_i \ne b_j \end{cases}$$

  where δ is a non-negative constant.
- C_{ij} (resp. R_{ij}) is the highest similarity among global alignments of $A[1..i]$ and $B[1..j]$ such that the last mapping is insertion (resp. deletion). C_{ij} and R_{ij} are computed by the following recurrence:

$$\begin{aligned} C_{0,j} = R_{i,0} &= -\infty \quad \text{for } 0 \le i \le n, 0 \le j \le m \\ C_{ij} &= \max\left\{H_{i-1,j} - g_1, C_{i-1,j} - \mu\right\} \quad \text{for } 1 \le i \le n,\ 1 \le j \le m \\ R_{ij} &= \max\left\{H_{i,j-1} - g_1, R_{i,j-1} - \mu\right\} \quad \text{for } 1 \le i \le n,\ 1 \le j \le m. \end{aligned} \tag{2}$$

Then the value H_{nm} is $SG(A, B)$ and it is computed in $O(nm)$ time.

2.2. Gap penalty models [8]

We defined the gap penalty g_k as $g_k = \gamma + k\mu$ where γ and μ are non-negative constants. This is called the *affine gap penalty model*, where γ is the gap initiation penalty and μ is the gap extension penalty. We define $g_0 = 0$. When there is no gap initiation penalty, i.e., $g_k = k\mu$, it is called the *linear gap penalty model*.

The problem we consider in this paper is as follows.

Problem 1. *Let A and B be two strings, and let A' and B' be run-length encoded strings of A and B, respectively. Given A' and B', compute $SG(A, B)$ with affine gap penalty.*

2.3. Black and white blocks [2]

We divide the H *table* into submatrices, which called "blocks". A *block* is a submatrix $H_{i_1..i_2, j_1..j_2}$ made up of two runs — one of A and one of B. Thus, by definition, the H *table* is divided into exactly $n'm'$ blocks where n' and m' are the run-length encoded lengths of A and B, respectively. The blocks are of two types: *black block*, corresponding to a pair of identical letters $a_{i_1} = b_{j_1}$, and *white block*, corresponding to a pair of distinct letters $a_{i_1} \ne b_{j_1}$.

Within one block, there exists only one kind of similarity score $s(a_i, b_j)$. In a black block, every a_i is equal to every b_j and thus $s(a_i, b_j)$ is always 1. In a white block, every a_i is different from every b_j and thus $s(a_i, b_j)$ is always $-\delta$.

2.4. Dependency of elements

The computation of similarity can be viewed as a path problem on a directed acyclic graph called an *alignment graph* [9]. See Fig. 1. At each position (i, j) for $0 \le i \le n$ and $0 \le j \le m$, there are three kinds of vertices: an *H-vertex*, a *C-vertex* and an *R-vertex*. An alignment graph has the following edges:

(1) h_1 : a horizontal edge from an H-vertex at (i, j) to an R-vertex at $(i, j+1)$. The edge weight $|h_1|$ is $-\gamma - \mu$.
(2) $\hat{h}_1$: a horizontal edge from an R-vertex at (i, j) to an R-vertex at $(i, j+1)$. $|\hat{h}_1| = -\mu$.
(3) v_1 : a vertical edge from an H-vertex at (i, j) to a C-vertex at $(i+1, j)$. $|v_1| = -\gamma - \mu$.

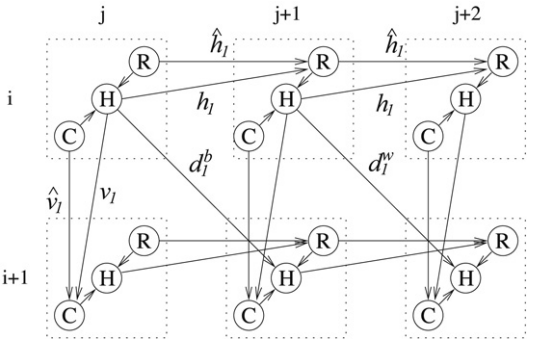

Fig. 1. An alignment graph for $a_{i+1} = g$ and $b_{j+1}b_{j+2} = gt$.

(4) $\hat{v}_1$: a vertical edge from a C-vertex at (i, j) to a C-vertex at $(i + 1, j)$. $|\hat{v}_1| = -\mu$.
(5) d_1 : a diagonal edge from an H-vertex at (i, j) to an H-vertex at $(i + 1, j + 1)$. There are two kinds of diagonal edges: d_1^b when $a_{i+1} = b_{j+1}$ and d_1^w when $a_{i+1} \neq b_{j+1}$. $|d_1^b| = 1$ and $|d_1^w| = -\delta$.
(6) Edges at (i, j) from an R-vertex to an H-vertex and from a C-vertex to an H-vertex. The edge weights are 0.

The edges from 1 to 4 are defined from recurrence (2) and the edges from 5 to 6 are defined from recurrence (1). Since R_{ij} is the maximum of $H_{i,j-1} - g_1$ and $R_{i,j-1} - \mu$ in recurrence (2), we define an edge h_1 from an H-vertex to an R-vertex with edge weight $-\gamma - \mu$ and define an edge $\hat{h}_1$ from an R-vertex to an R-vertex with $-\mu$. The other edges are defined similarly.

We can define a *path* $\langle\cdot\rangle$ from a vertex to a vertex. A horizontal path $\langle\hat{h}_i\rangle$ for $i > 1$ is defined as i consecutive $\hat{h}_1$ edges, i.e., $\langle\hat{h}_1 \ldots \hat{h}_1\rangle$ and a horizontal path $\langle h_i\rangle$ is defined as $\langle h_1\hat{h}_{i-1}\rangle$. Vertical paths $\langle\hat{v}_i\rangle$ and $\langle v_i\rangle$ are defined similarly. A diagonal path $\langle d_i\rangle$ is defined as i consecutive d_1 edges. A path P from (k, l) to (i, j) is a sequence of edges from a vertex at (k, l) to a vertex at (i, j). For example, $\langle h_2 d_1 v_1\rangle$ is a path from an H-vertex at (i, j) to a C-vertex (or an H-vertex) at $(i + 2, j + 3)$. Let $|\langle\cdot\rangle|$ denote a path weight of $\langle\cdot\rangle$ which is the sum of all edge weights in the path. For example, the path weight of $\langle h_2 d_1^w v_1\rangle$ is $|\langle h_2 d_1^w v_1\rangle| = -\gamma - 2\mu - \delta - \gamma - \mu$.

We now describe various properties of paths which will be used in Section 3. Let $\leftrightarrow$ denote an equivalence relation between two paths which means that the numbers of each kind of edges are the same on both sides. This implies that the path weights of the two paths are the same. Note that $\leftrightarrow$ is symmetric.

We can merge two paths into one or divide a path into two. For example, if $\langle d_a h_b\rangle\langle\hat{h}_c v_d\rangle$ is a sequence of paths from (k, l) to (i, j), then $\langle d_a h_b\hat{h}_c v_d\rangle$ is also a path from (k, l) to (i, j), i.e., $\langle d_a h_b\rangle\langle\hat{h}_c v_d\rangle \leftrightarrow \langle d_a h_b\hat{h}_c v_d\rangle$. Also $\langle d_a h_b\hat{h}_c v_d\rangle$ can be divided into $\langle d_a h_b\rangle\langle\hat{h}_c v_d\rangle$, i.e., $\langle d_a h_b\hat{h}_c v_d\rangle \leftrightarrow \langle d_a h_b\rangle\langle\hat{h}_c v_d\rangle$.

Fact 1. $\langle\alpha\rangle\langle\beta\rangle \leftrightarrow \langle\alpha\beta\rangle$ *where* α *and* β *are sequences of edges.*

By definition of a path, $\langle d_a h_b\hat{h}_c v_d\rangle \leftrightarrow \langle d_a h_{b+c} v_d\rangle$. However, for the following cases, the path weights are changed:

Fact 2. $|\langle h_a h_b\rangle| \leq |\langle h_{a+b}\rangle|$ *and* $|\langle v_a v_b\rangle| \leq |\langle v_{a+b}\rangle|$.

We can exchange the order of two adjacent edges in a path. If $\langle h_a v_b d_c\rangle$ is a path from (k, l) to (i, j), then $\langle v_b h_a d_c\rangle$ is also a path from (k, l) to (i, j) and the path weights are the same.

Fact 3. $\langle h_a v_b\rangle \leftrightarrow \langle v_b h_a\rangle$.

However, an exchange of edge d can cause the change of a path weight because d depends on the match/mismatch of the position. Since $|d_1^w| < |d_1^b|$, we get the following fact.

Fact 4. $|\langle v_a d_b^w\rangle| \leq |\langle d_b v_a\rangle|$ *and* $|\langle v_a d_b^b\rangle| \geq |\langle d_b v_a\rangle|$.

For a maximum path in one block, of course, the order exchange of d does not change the path weight because there is only one kind of edge d in one block.

Fact 5. $|\langle\alpha d_b^w\rangle| = |\langle d_b^w\alpha\rangle|$ *and* $|\langle\alpha d_b^b\rangle| = |\langle d_b^b\alpha\rangle|$ *within one block.*

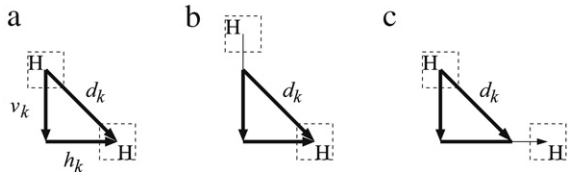

Fig. 2. Three cases for edge selection. $|\langle d_k^w \rangle|$ is compared with (a) $|\langle v_k h_k \rangle|$, (b) $|\langle \hat{v}_k h_k \rangle|$, (c) $|\langle v_k \hat{h}_k \rangle|$.

We also define a *maximum path* from (k, l) to (i, j) which is a path that has the maximum path weight among all paths from (k, l) to (i, j). A maximum path from an H-vertex to another H-vertex will be called an *HH-mp*. Similarly, we will use a *CC-mp*, an *RR-mp*, an *HC-mp*, etc. Each maximum path has some restrictions: An Hx-mp cannot start with $\hat{v}$ or $\hat{h}$ (x is a do-not-care symbol). A Cx-mp and an Rx-mp must start with $\hat{v}$ and $\hat{h}$, respectively. An xC-mp must end with v or $\hat{v}$ and an xR-mp must end with h or $\hat{h}$.

From recurrence (1), we can get a relation between H_{ij} and its previously defined entries.

Lemma 6. *Let P be an HH-mp from (k, l) to (i, j). Then $H_{ij} \geq H_{kl} + |P|$.*

Note that the symmetric versions of Lemma 6 hold for a *CC*-mp, an *RR*-mp, an *HC*-mp, etc.

Now we consider a maximum path in one block. By Facts 2, 3 and 5, we get the following fact.

Fact 7. *Every maximum path in one block is represented as a permutation of at most one d_i, at most one h_j and at most one v_k (except CC-mp and RR-mp which may have additionally one $\hat{v}_p$ and one $\hat{h}_q$, respectively).*

The number of diagonal edges in a maximum path depends on the weight of d and those of v and h. For a black block, $|\langle d_k^b \rangle| = k > 0$ and $|\langle h_k \rangle| = |\langle v_k \rangle| \leq 0$ (also $|\langle \hat{h}_k \rangle| = |\langle \hat{v}_k \rangle| \leq 0$). Thus we get the following fact.

Fact 8. *In a black block, the number of diagonal edges in a maximum path must be maximized.*

For a white block, Fact 8 does not hold because the similarity score for mismatch, $-\delta$, is also less than or equal to 0. Instead, we show that for an *HH*-mp from (i, j) to $(i + k, j + k)$, the number of diagonal edges in a maximum path is either k or 0. By Fact 7, a maximum path from (i, j) to $(i + k, j + k)$ for $0 \leq t \leq k$ is $\langle v_{k-t} d_t^w h_{k-t} \rangle$. Then the path weight is $|\langle v_{k-t} d_t^w h_{k-t} \rangle| = -2g_{k-t} - t\delta = -2\gamma\lceil (k-t)/k \rceil - 2k\mu + (2\mu - \delta)t$ since $g_{k-t} = -\gamma - (k-t)\mu$ if $t < k$; it is 0 if $t = k$. The term $-2\gamma\lceil (k-t)/k \rceil$ has a maximum value when $t = k$ and the term $(2\mu - \delta)t$ has a maximum value when $t = k$ for $2\mu - \delta \geq 0$ and when $t = 0$ for $2\mu - \delta < 0$. By these, the path has the maximum weight only when $t = k$ or $t = 0$, i.e., maximum path $\langle v_{k-t} d_t^w h_{k-t} \rangle$ is either $\langle d_k^w \rangle$ when $t = k$ or $\langle v_k h_k \rangle$ when $t = 0$. Thus we get the following fact.

Fact 9. *In a white block, an HH-mp from (i, j) to $(i + k, j + k)$ is either $\langle d_k^w \rangle$ or $\langle v_k h_k \rangle$. In addition, an HH-mp from (i, j) to $(i + k + s, j + k)$ for $s > 0$ is either $\langle v_s d_k^w \rangle$ or $\langle v_{s+k} h_k \rangle$. An HH-mp from (i, j) to $(i + k, j + k + s)$ is either $\langle d_k^w h_s \rangle$ or $\langle v_k h_{k+s} \rangle$. (See Fig. 2.)*

Note that the symmetric versions of Fact 9 hold for a *CH*-mp, an *RH*-mp, a *CC*-mp, etc.

3. Algorithm

In this section we present an algorithm that computes the similarity between two run-length encoded strings with affine gap penalty.

The outline of the algorithm is the same as that for the LCS [3], the Levenshtein distance [2,11] and the weighted edit distance [5,12]. Given two run-length encoded strings A' and B', we compute blocks from left to right and from top to bottom. For each block, we compute the bottom row from left to right and the rightmost column from top to bottom. See Fig. 3.

Given a block $H_{i+1..i+p,j+1..j+q}$, our goal is to compute the value of $C_{i+p,j+l}$, $R_{i+p,j+l}$ and $H_{i+p,j+l}$ for $1 \leq l \leq q$ (bottom row) and $C_{i+k,j+q}$, $R_{i+k,j+q}$ and $H_{i+k,j+q}$ for $1 \leq k \leq p$ (rightmost column) in $O(p + q)$ time using $C_{i+k,j}$, $R_{i+k,j}$ and $H_{i+k,j}$ for $0 \leq k \leq p$ (leftmost column) and $C_{i,j+l}$, $R_{i,j+l}$ and $H_{i,j+l}$ for $0 \leq l \leq q$ (top row).

We present two algorithms, one for a white block and another for a black block. For each block, we first present how to compute the values of C and R, and then show how to compute the values of H.

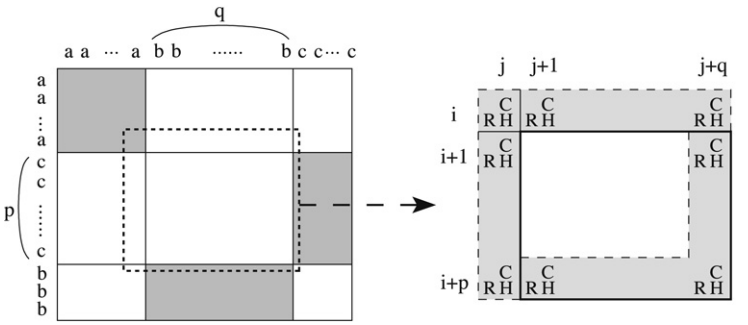

Fig. 3. *H table* for $a^r c^p b^t$ and $a^s b^q c^u$ is divided into 9 blocks which consist of 3 black blocks and 6 white blocks. For one of the white blocks, $H_{i+1..i+p,j+1..j+q}$, we only need to compute $H_{i+p,j+1..j+q}$ and $H_{i+1..i+p,j+q}$ from $H_{i..i+p,j}$ and $H_{i,j..j+q}$.

3.1. White blocks

We give an algorithm for a white block. We only show how to compute the values of the elements on the bottom row of the block. Computing the elements on the rightmost column is done similarly.

3.1.1. Computing $C_{i+p,j+l}$

To compute the value of $C_{i+p,j+l}$ for $1 \le l \le q$, we need $R_{i+k,j}$ for $1 \le k \le p$, $C_{i,j+s}$ for $1 \le s \le l$, $H_{i+k,j}$ for $1 \le k \le p$, and $H_{i,j+s}$ for $0 \le s \le l$. Since there are various ways from each element to $C_{i+p,j+l}$, we give some lemmas to select *essential paths*, i.e., the paths that must be considered to compute $C_{i+p,j+l}$.

Lemma 10. *Let $H_{i+k,j+l}$ for $1 \le k \le p$ be an element within a white block and P_1 be a CH-mp from $(i, j+l-s)$ to $(i+k, j+l)$ for $0 \le s < l$. Then, there exists an element $C_{i,j+l}$ such that $C_{i,j+l-s}+|P_1| \le C_{i,j+l}+|P_2|$ where P_2 is a CH-mp from $(i, j+l)$ to $(i+k, j+l)$ or there exists $H_{i,j+l-t}$ for $0 \le t < s$ such that $C_{i,j+l-s}+|P_1| \le H_{i,j+l-t}+|P_3|$ where P_3 is an HH-mp from $(i, j+l-t)$ to $(i+k, j+l)$.*

Proof. In a white block, the *CH*-mp P_1 is $\langle \hat{v}_l h_s \rangle$, $\langle \hat{v}_{l-s} d_s^w \rangle$, or $\langle \hat{v}_1 d_{k-1}^w h_{s-k+1} \rangle$ by Fact 9. We prove the lemma in three cases.

(i) $P_1 = \langle \hat{v}_l h_s \rangle$: See Fig. 4(a). By recurrence (2), there exists $H_{i-u,j+l-s}$ such that $C_{i,j+l-s} = H_{i-u,j+l-s} - g_u$, i.e., $C_{i,j+l-s} = H_{i-u,j+l-s} + |\langle v_u \rangle|$. By Facts 1 and 3, $\langle v_u \rangle \langle \hat{v}_k h_s \rangle \leftrightarrow \langle v_u \hat{v}_k h_s \rangle \leftrightarrow \langle h_s v_u \hat{v}_k \rangle \leftrightarrow \langle h_s v_u \rangle \langle \hat{v}_k \rangle$. Now we consider the element $C_{i,j+l}$. Then $C_{i,j+l} \ge H_{i-u,j+l-s} + |\langle h_s v_u \rangle|$ by Lemma 6 and the *CH*-mp P_2 from $(i, j+l)$ to $(i+k, j+l)$ is $\langle \hat{v}_k \rangle$. Thus,

$$\begin{aligned} C_{i,j+l-s} + |P_1| &= H_{i-u,j+l-s} + |\langle v_u \rangle| + |\langle \hat{v}_k h_s \rangle| \\ &= H_{i-u,j+l-s} + |\langle h_s v_u \rangle| + |\langle \hat{v}_k \rangle| \\ &\le C_{i,j+l} + |\langle \hat{v}_k \rangle| = C_{i,j+l} + |P_2| \end{aligned}$$

and the lemma holds.

(ii) $P_1 = \langle \hat{v}_{k-s} d_s^w \rangle$: As in (i), $C_{i,j+l-s} = H_{i-u,j+l-s} + |\langle v_u \rangle|$. By Facts 1 and 4, $\langle v_u \rangle \langle \hat{v}_{k-s} d_s^w \rangle \leftrightarrow \langle v_{u+k-s} d_s^w \rangle$ and $|\langle v_{u+k-s} d_s^w \rangle| \le |\langle d_s v_{u+k-s} \rangle|$. Suppose that $u > s$. See Fig. 4(b). We consider the element $C_{i,j+l}$. Then $C_{i,j+l} \ge H_{i-u,j+l-s} + |\langle d_s v_{u-s} \rangle|$ by Lemma 6 and the *CH*-mp P_2 from $(i, j+l)$ to $(i+k, j+l)$ is $\langle \hat{v}_k \rangle$. Thus,

$$\begin{aligned} C_{i,j+l-s} + |P_1| &= H_{i-u,j+l-s} + |\langle v_u \rangle| + |\langle \hat{v}_{k-s} d_s^w \rangle| \\ &\le H_{i-u,j+l-s} + |\langle d_s v_{u-s} \rangle| + |\langle \hat{v}_k \rangle| \\ &\le C_{i,j+l} + |\langle \hat{v}_k \rangle| = C_{i,j+l} + |P_2|. \end{aligned}$$

Suppose that $u \le s$. See Fig. 4(c). We consider the element $H_{i,j+l-t}$ where $t = s - u$. Then $H_{i,j+l-t} \ge H_{i-u,j+l-s} + |\langle d_u \rangle|$ by Lemma 6 and the *HH*-mp P_3 from $(i, j+l-t)$ to $(i+k, j+l)$ is $\langle d_t v_{k-t} \rangle$. Thus, $C_{i,j+l-s} + |P_1| \le H_{i-u,j+l-s} + |\langle d_u \rangle| + |\langle d_t v_{k-t} \rangle| \le H_{i,j+l-t} + |P_3|$ and the lemma holds.

(iii) $P_1 = \langle \hat{v}_1 d_{k-1}^w h_{s-k+1} \rangle$: See Fig. 4(d). As in (i), $C_{i,j+l-s} = H_{i-u,j+l-s} + |\langle v_u \rangle|$. By Facts 1 and 5, $\langle v_u \rangle \langle \hat{v}_1 d_{k-1}^w h_{s-k+1} \rangle \leftrightarrow \langle v_u \hat{v}_1 d_{k-1}^w h_{s-k+1} \rangle \leftrightarrow \langle h_{s-k+1} v_u \hat{v}_1 d_{k-1}^w \rangle$. A *CH*-mp from $(i, j+l-k+1)$ to $(i+k, j+l)$

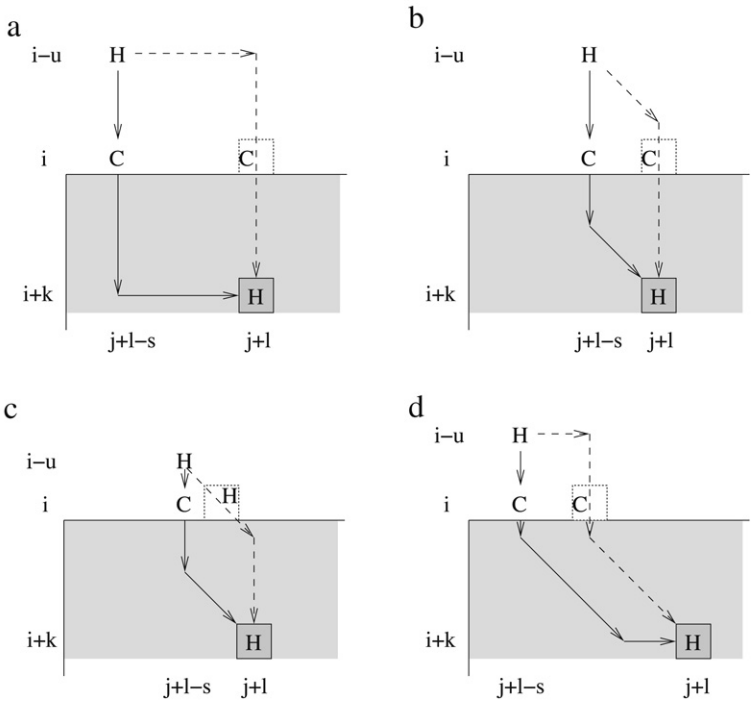

Fig. 4. Proof of Lemma 10.

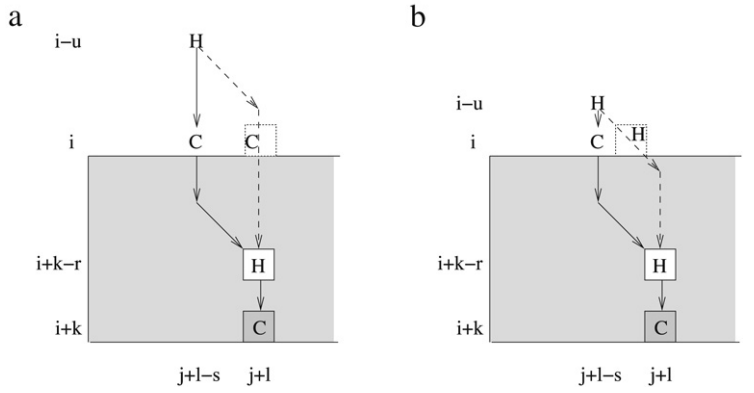

Fig. 5. Proof of Lemma 11.

is $\langle \hat{v}_1 d_{k-1} \rangle$. Thus,

$$\begin{aligned} C_{i,j+l-s} + |P_1| &= H_{i-u,j+l-s} + |\langle h_{s-k+1} v_u \rangle| + |\langle \hat{v}_1 d^w_{k-1} \rangle| \\ &= C_{i,j+l-k+1} + |\langle \hat{v}_1 d^w_{k-1} \rangle| \end{aligned}$$

and then it becomes case (ii). Hence the lemma holds. □

Lemma 11. *Let P_1 be a CC-mp from $(i, j+l-s)$ to $(i+p, j+l)$ for $0 \le s < l$. Then, there exists an element $C_{i,j+l}$ such that $C_{i,j+l-s} + |P_1| \le C_{i,j+l} + |P_2|$ where P_2 is a CC-mp from $(i, j+l)$ to $(i+p, j+l)$ or there exists an element $H_{i,j+l-t}$ for $0 \le t < s$ such that $C_{i,j+l-s} + |P_1| \le H_{i,j+l-t} + |P_3|$ where P_3 is an HC-mp from $(i, j+l-t)$ to $(i+p, j+l)$.*

Proof. See Fig. 5. Because P_1 is a *CC*-mp from $(i, j+l-s)$ to $(i+p, j+l)$, $P_1 = P'_1 + \langle v_r \rangle$ for $1 \le r < k$ where P'_1 is a *CH*-mp from $(i, j+l-s)$ to $(i+p-r, j+l)$, i.e., $C_{i,j+l-s} + |P'_1| = H_{i+p-r,j+l}$ and $H_{i+p-r,j+l} + |\langle v_r \rangle| = C_{i+p,j+l}$. By

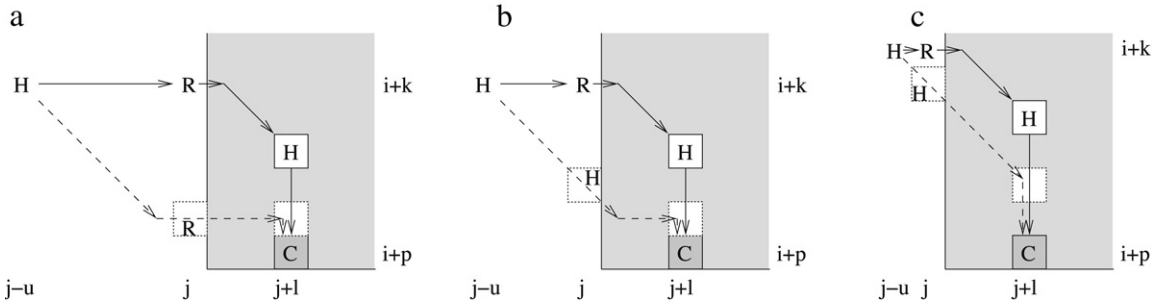

Fig. 6. Proof of Lemma 12.

Lemma 10, there exists an element $C_{i,j+l}$ such that $C_{i,j+l-s}+|P_1'| \le C_{i,j+l}+|P_2'|$ or there exists an element $H_{i,j+l-t}$ such that $C_{i,j+l-s}+|P_1'| \le H_{i,j+l-t}+|P_3'|$. Thus, $C_{i,j+l-s}+|P_1| = C_{i,j+l-s}+|P_1'|+\langle v_r\rangle \le C_{i,j+l}+|P_2'|+\langle v_r\rangle \le C_{i,j+l}+|P_2|$ or $C_{i,j+l-s}+|P_1| \le H_{i,j+l-t}+|P_3'|+\langle v_r\rangle \le H_{i,j+l-t}+|P_3|$. □

Lemma 12. *Let P_1 be an RC-mp from $(i+k, j)$ to $(i+p, j+l)$ for $1 \le k \le p-1$. If $-\delta > -2\mu$, there exists an element $R_{i+p-1,j}$ such that $R_{i+k,j}+|P_1| \le R_{i+p-1,j}+|P_2|$ where P_2 is an RC-mp from $(i+p-1, j)$ to $(i+p, j+l)$ or there exists $H_{i+t,j}$ for $k < t \le p-1$ such that $R_{i+k,j}+|P_1| \le H_{i+t,j}+|P_3|$ where P_3 is an HC-mp from $(i+t, j)$ to $(i+p, j+l)$.*

Proof. Since $-\delta > -2\mu$, the *RC*-mp $P_1 = \langle \hat{h}_{l-a} d_a^w v_{p-k-a}\rangle$ where $a = \min\{l-1, p-k-1\}$. By recurrence (2), there exists $H_{i+k,j-u}$ for $u > 0$ such that $R_{i+k,j} = H_{i+k,j-u} + |\langle h_u\rangle|$. Then, $R_{i+k,j}+|P_1| = H_{i+k,j-u} + |\langle h_u \hat{h}_{l-a} d_a^w v_{p-k-a}\rangle|$. By Fact 4, $|\langle h_u \hat{h}_{l-a} d_a^w v_{p-k-a}\rangle| \le |\langle d_a h_{u+l-a} v_{p-k-a}\rangle|$. Since $-\delta > -2\mu$, we can maximize the number of diagonal edges without considering the type of a block. Let $b = \min\{u+l-a, p-k-a-1\}$. Then, $|\langle d_a h_{u+l-a} v_{p-k-a}\rangle| < |\langle d_{a+b} h_{u+l-a-b} v_{p-k-a-b}\rangle|$.

Suppose $a+b < u$. See Fig. 6(a). Now we consider the element $R_{i+p-1,j}$. Then $R_{i+p-1,j} \ge H_{i+k,j-u} + |\langle d_{a+b} h_{u-a-b}\rangle|$ by Lemma 6 and the *RC*-mp P_2 from $(i+p-1, j)$ to $(i+p, j+l)$ is $\langle \hat{h}_l v_1\rangle$. Thus, $H_{i+k,j-u} + |\langle d_{a+b} h_{u+l-a-b} v_{p-k-a-b}\rangle| \le R_{i+p-1,j} + |\langle \hat{h}_l v_{p-k-a-b}\rangle|$. Since $i+k+a+b = i+p-1$, $p-k-a-b = 1$ and thus $R_{i+k,j}+|P_1| \le R_{i+p-1,j}+|P_2|$ from above all.

Suppose $a+b \ge u$. See Fig. 6(b) and Fig. 6(c). We note that $k+a+b \le p-1$. We consider $H_{i+t,j}$ where $t = k+u \le p-1$. By Lemma 6, $H_{i+t,j} \ge H_{i+k,j-u} + |\langle d_u\rangle|$ and thus $H_{i+k,j-u} + |\langle d_{a+b} h_{u+l-a-b} v_{p-k-a-b}\rangle| \le H_{i+t,j} + |\langle d_{a+b-u} h_{u+l-a-b} v_{p-k-a-b}\rangle|$. Since $\langle d_{a+b-u} h_{u+l-a-b} v_{p-k-a-b}\rangle$ is a path from $(i+t, j)$ to $(i+p, j+l)$, the weight of this path is not greater than that of the *HC*-mp P_3. Hence $R_{i+k,j}+|P_1| \le H_{i+k+u,j}+|P_3|$ from above all. □

If $-\delta \le -2\mu$, the *RC*-mp from $(i+k, j)$ to $(i+p, j+l)$ for $1 \le k \le p-1$ is $\langle \hat{h}_l v_{p-k}\rangle$ and it is an essential path for every $1 \le k \le p-1$.

Lemma 13. *Let P_1 be an HC-mp from $(i, j+l-s)$ to $(i+p, j+l)$ for $0 \le s \le l$. Then, there exists an element $H_{i,j+l-t}$ for $0 \le t \le \min\{l, p-1\}$ such that $H_{i,j+l-s}+|P_1| \le H_{i,j+l-t}+|P_2|$ where P_2 is an HC-mp from $(i, j+l-t)$ to $(i+p, j+l)$.*

Proof. If $s < p$, then let t be s and it is done. Now, we consider that $s \ge p$. In a white block, the *HC*-mp P_1 is $\langle v_{p-1} h_s v_1\rangle$ or $\langle d_{p-1} h_{s-p+1} v_1\rangle$ by Fact 9. We prove the lemma in two cases.

(i) $P_1 = \langle v_{p-1} h_s v_1\rangle$: See Fig. 7(a). By Facts 3 and 2, $\langle v_{p-1} h_s v_1\rangle \leftrightarrow \langle h_s v_{p-1} v_1\rangle$ and $|\langle h_s v_{p-1} v_1\rangle| \le |\langle h_s v_p\rangle|$. Now we consider the element $H_{i,j+l}$. Then $H_{i,j+l} \ge H_{i,j+l-s}+|\langle h_s\rangle|$ by Lemma 6 and the *HC*-mp P_2 from $(i, j+l)$ to $(i+p, j+l)$ is $\langle v_p\rangle$. Thus, $H_{i,j+l-s}+|P_1| \le H_{i,j+l-s}+|\langle h_s v_p\rangle| \le H_{i,j+l}+|\langle v_p\rangle| = H_{i,j+l}+|P_2|$ and the lemma holds.

(ii) $P_1 = \langle d_{p-1} h_{s-p+1} v_1\rangle$: See Fig. 7(b). By Fact 5, $|\langle d_{p-1} h_{s-p+1} v_1\rangle| = |\langle h_{s-p+1} d_{p-1} v_1\rangle|$. Now we consider the element $H_{i,j+l-p+1}$. Then $H_{i,j+l-p+1} \ge H_{i,j+l-s}+|\langle h_{s-p+1}\rangle|$ by Lemma 6 and the *HC*-mp P_2 from $(i, j+l-p+1)$ to $(i+p, j+l)$ is $\langle d_{p-1} v_1\rangle$. Thus, $H_{i,j+l-s}+|P_1| = H_{i,j+l-s}+|\langle h_{s-p+1} d_{p-1} v_1\rangle| \le H_{i,j+l-p+1}+|\langle d_{p-1} v_1\rangle| = H_{i,j+l-p+1}+|P_2|$ and the lemma holds. □

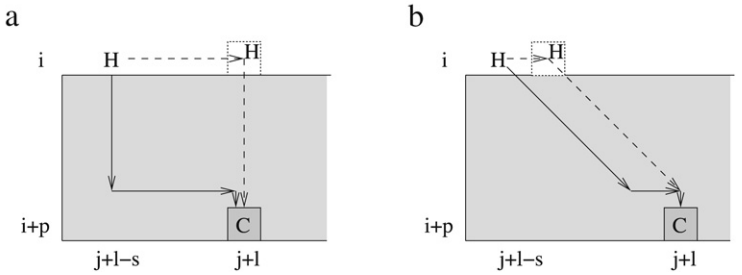

Fig. 7. Proof of Lemma 13.

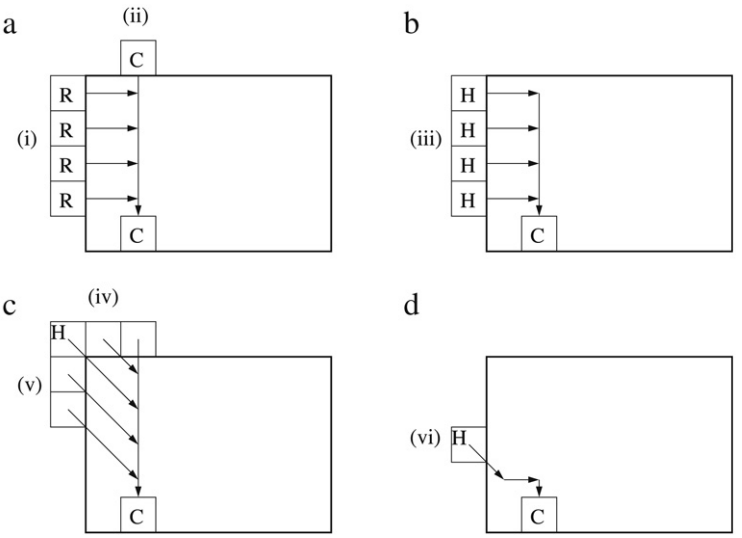

Fig. 8. Computing $C_{i+p,j+l}$ in a white block. (a) formulas (i) and (ii), (b) formula (iii), (c) formulas (iv) and (v), (d) formula (vi).

By the lemmas above, we can select essential paths from the top row of C, the leftmost column of R, and the top row of H of the block to $C_{i+p,j+l}$. The maximum paths from the leftmost column of H to $C_{i+p,j+l}$, i.e., the HC-mps from $(i+k, j)$ to $(i+p, j+l)$ for $1 \le k \le p-1$, are all essential paths. From these, we derive that the value of $C_{i+p,j+l}$ is the maximum of the following. See Fig. 8:

(i) $\max_{1\le s\le p-1}\{R_{i+s,j} - g_{p-s}\} - l\mu$
(ii) $C_{i,j+l} - p\mu$
(iii) $\max_{1\le s\le p-1}\{H_{i+s,j} - g_{p-s}\} - g_l$
(iv) $\max_{0\le s\le \min\{l,p-1\}}\{H_{i,j+l-s} - s\delta - g_{p-s}\}$
(v) $\max_{1\le s\le p-1-l}\{H_{i+s,j} - g_{p-s-l}\} - l\delta$ when $l < p-1$
(vi) $\max_{1\le s\le \min\{l-1,p-2\}}\{H_{i+p-1-s,j} - g_{l-s} - s\delta\} - g_1$ when $l \ge 2$.

The value of each formula can be computed in $O(p)$ time (of course, (ii) in constant time) and the maximum of them is computed in constant time. Thus we need $O(p)$ time to compute the value of $C_{i+p,j+l}$.

Computing all the values of C of the bottom row needs $O(pq)$ time using the above result. However, since we compute the bottom row from left to right, i.e., l is increased from 1 to q, we can reduce the time complexity to $O(p+q)$ using such properties of the recurrences that two adjacent entries are very similar.

First, consider (i) $\max_{1\le s\le p-1}\{R_{i+s,j} - g_{p-s}\} - l\mu$ and (iii) $\max_{1\le s\le p-1}\{H_{i+s,j} - g_{p-s}\} - g_l$. The index s of the maximum value in (i) and that in (iii) do not depend on l. Hence we compute (i) and (iii) for $l = 1$ in $O(p)$ time and then get the maximum value for $l \ge 2$ in constant time by adding $-(l-1)\mu$.

Second, consider (iv) $\max_{0\le s\le \min\{l,p-1\}}\{H_{i,j+l-s} - s\delta - g_{p-s}\}$. The range of the column index for H in (iv) is j to $j+l$ for $1 \le l < p$ and $j+l-p+1$ to $j+l$ for $l \ge p$. As l increases, the range is increased by one till $l < p$ and then the position of the range is shifted to the right by one. See Fig. 9(a) and Fig. 9(c). Each time l increases, value

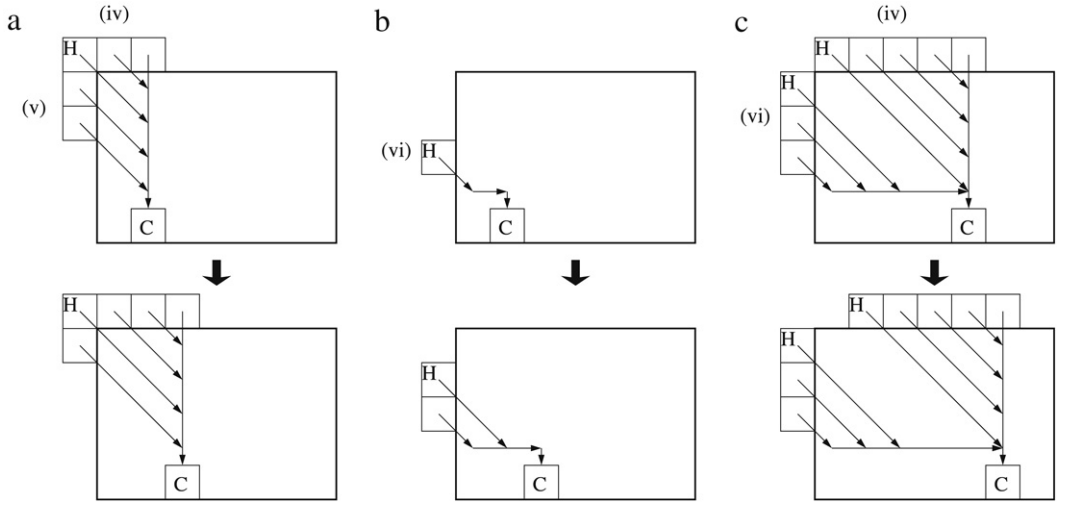

Fig. 9. The changes from $C_{i+p,j+l}$ to $C_{i+p,j+l+1}$ when (a) (b) $l < p-1$ and (c) $l \geq p-1$.

$-\delta+\mu$ is added to all the rest of the elements. It is almost the same as the recurrence for C in Case 2 of [10]. Thus, using MQUEUE [10], we can get the maximum value in amortized constant time. We can use a deque with heap order [6] to get worst-case constant time.

Third, consider (v) $\max_{1\leq s\leq p-1-l}\{H_{i+s,j}-g_{p-s-l}\}-l\delta$ when $l<p-1$. The range of the row index for H in (v) is $i+1$ to $i+p-1-l$ for $1\leq l<p-1$. That is, the range is decreased by one till $l<p-1$. See Fig. 9(a). Hence we make a stack with heap order for $l=1$ in $O(p)$ time and then get the maximum value for $l\geq 2$ one-by-one in constant time by popping one element, getting the maximum value of the stack and adding $(l-1)(-\delta+\mu)$ to it.

Last, consider (vi) $\max_{1\leq s\leq \min\{l-1,p-2\}}\{H_{i+p-1-s,j}-g_{l-s}-s\delta\}-g_1$ when $l\geq 2$. The range of the row index for H in (vi) is $i+p-l$ to $i+p-2$ for $2\leq l<p-1$ and $i+1$ to $i+p-2$ for $l\geq p-1$. As l increases, the range is increased by one till $l<p-1$ and then the index s of the maximum value does not depend on l for $l\geq p-1$. See Fig. 9(b) and Fig. 9(c). Thus, we can get the maximum value for $l=2$ in constant time and then get the maximum value till $l<p-1$ one-by-one in constant time by adding $-\mu$ to the previous maximum value and comparing it with a new element. We also get the maximum value for $l\geq p-1$ in constant time by adding $-(l-p+2)\mu$ to the maximum value for $l=p-2$.

From above all, we compute (i) and (iii) in $O(p+q)$ time, (ii), (iv) and (vi) in $O(q)$ time, and (v) in $O(p)$ time. Therefore, we compute $C_{i+p,j+l}$ for $1\leq l\leq q$ in $O(p+q)$ time.

3.1.2. Computing $R_{i+p,j+l}$

To compute the value of $R_{i+p,j+l}$ for $1\leq l\leq q$, we need $R_{i+p,j+l-1}$ and $H_{i+p,j+l-1}$ by recurrence (2). Since we know the values of $R_{i+p,j}$ and $H_{i+p,j}$ and we compute the bottom row of R from left to right, we have no problem to compute $R_{i+p,j+l}$ and it takes $O(1)$ time. Therefore, we compute all the values of R of the bottom row in $O(q)$ time.

3.1.3. Computing $H_{i+p,j+l}$

To compute the value of $H_{i+p,j+l}$ for $1\leq l\leq q$, we need $C_{i+p,j+l}$, $R_{i+p,j+l}$ and $H_{i+p-1,j+l-1}$. Since we know the values of $C_{i+p,j+l}$ and $R_{i+p,j+l}$, we only need to compute the diagonal incoming value.

Lemma 14. *Let P_1 be an RH-mp from $(i+k,j)$ to $(i+p-1,j+l-1)$ for $1\leq k\leq p-1$. Then, $R_{i+k,j}+|P_1|+|\langle d_1^w\rangle|\leq R_{i+p,j+l}$.*

Proof. We denote the *RH*-mp P_1 by $\langle\hat{h}_r\alpha\rangle$ for $r\geq 1$ where α is a sequence of edges which starts with d^w or v. $\langle d_1^w\rangle$ is a diagonal path from $(i+p-1,j+l-1)$ to $(i+p,j+l)$. By recurrence (2), there exists $H_{i+k,j-u}$ such that $R_{i+k,j}=H_{i+k,j-u}+|\langle h_u\rangle|$. See Fig. 10. By Facts 1 and 4, $\langle h_u\rangle\langle\hat{h}_r\alpha\rangle\langle d_1^w\rangle\leftrightarrow\langle h_{u+r}\alpha d_1^w\rangle$ and $|\langle h_{u+r}\alpha d_1^w\rangle|\leq|\langle\alpha d_1 h_{u+r}\rangle|$. Let P_2 be an *HR*-mp from $(i+k,j-u)$ to $(i+p,j+l)$. Since $\langle\alpha d_1 h_{u+r}\rangle$ is

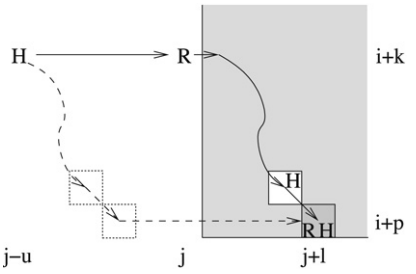

Fig. 10. Proof of Lemma 14.

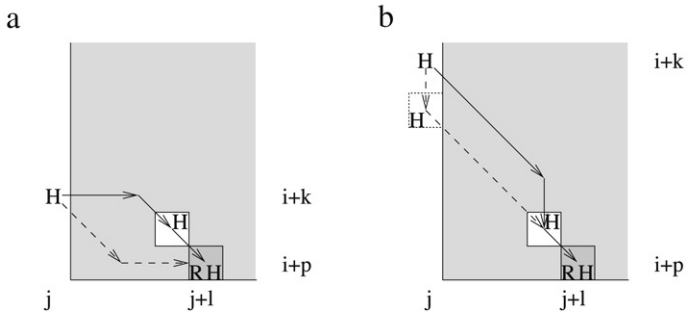

Fig. 11. Proof of Lemma 16.

a path from $(i+k, j-u)$ to $(i+p, j+l)$, the weight of this path is not greater than that of P_2. By Lemma 6, $R_{i+p,j+l} \geq H_{i+k,j-u} + |P_2|$. Thus,

$$\begin{aligned} R_{i+k,j} + |P_1| + |\langle d_1^w \rangle| &= H_{i+k,j-u} + |\langle h_{u+r} \alpha d_1^w \rangle| \\ &\leq H_{i+k,j-u} + |\langle \alpha d_1 h_{u+r} \rangle| \\ &\leq H_{i+k,j-u} + |P_2| \\ &\leq R_{i+p,j+l} \end{aligned}$$

and the lemma holds. □

Lemma 15. *Let P_1 be a CH-mp from $(i, j+s)$ to $(i+p-1, j+l-1)$ for $1 \leq s \leq l-1$. Then, $C_{i,j+s} + |P_1| + |\langle d_1^w \rangle| \leq C_{i+p,j+l}$.*

Proof. Similar to the proof of Lemma 14. □

Lemma 16. *Let P_1 be an HH-mp from $(i+k, j)$ to $(i+p-1, j+l-1)$ for $1 \leq k \leq p-1$. Then, $H_{i+k,j} + |P_1| + |\langle d_1^w \rangle| \leq R_{i+p,j+l}$ or $H_{i+k,j} + |P_1| + |\langle d_1^w \rangle| \leq H_{i+p-l,j} + |\langle d_l^w \rangle|$ when $l \leq p$.*

Proof. We prove the lemma in three cases.

(i) $i+k > i+p-l$: See Fig. 11(a). Since $0 \leq p-1-k < l-1$, the *HH*-mp P_1 is $\langle h_{l-1} v_{p-1-k} \rangle$ or $\langle h_{l-1-p+1+k} d_{p-1-k} \rangle$. Without loss of generality, we denote P_1 by $\langle h_r \alpha \rangle$ for $r \geq 1$ where α is a sequence of edges. $\langle d_1^w \rangle$ is a diagonal path from $(i+p-1, j+l-1)$ to $(i+p, j+l)$. By Facts 1 and 5, $\langle h_r \alpha \rangle \langle d_1^w \rangle \leftrightarrow \langle h_r \alpha d_1^w \rangle$ and $|\langle h_r \alpha d_1^w \rangle| = |\langle \alpha d_1^w h_r \rangle|$. Let P_2 be an *HR*-mp from $(i+k, j)$ to $(i+p, j+l)$. Since $\langle \alpha d_1^w h_r \rangle$ is a path from $(i+k, j)$ to $(i+p, j+l)$, the weight of this path is not greater than that of P_2. By Lemma 6, $R_{i+p,j+l} \geq H_{i+k,j} + |P_2|$. Thus, $H_{i+k,j} + |P_1| + |\langle d_1^w \rangle| = H_{i+k,j} + |\langle \alpha d_1^w h_r \rangle| \leq H_{i+k,j} + |P_2| \leq R_{i+p,j+l}$ and the lemma holds.

(ii) $i+k < i+p-l$: Since $p-1-k > l-1$, the *HH*-mp P_1 is $\langle h_{l-1} v_{p-1-k} \rangle$ or $\langle d_{l-1}^w v_{p-1-k-l+1} \rangle$. If $P_1 = \langle h_{l-1} v_{p-1-k} \rangle$, it becomes case (i). Now we consider $P_1 = \langle d_{l-1}^w v_{p-1-k-l+1} \rangle$. See Fig. 11(b). By Facts 1 and 5, $\langle d_{l-1}^w v_{p-k-l} \rangle \langle d_1^w \rangle \leftrightarrow \langle d_{l-1}^w v_{p-k-l} d_1^w \rangle$ and $|\langle d_{l-1}^w v_{p-k-l} d_1^w \rangle| = |\langle v_{p-k-l} d_l^w \rangle|$. By Lemma 6, $H_{i+p-l,j} \geq H_{i+k,j} + |\langle v_{p-l-k} \rangle|$. Thus, $H_{i+k,j} + |P_1| + |\langle d_1^w \rangle| = H_{i+k,j} + |\langle v_{p-k-l} d_l^w \rangle| \leq H_{i+p-l,j} + |\langle d_l^w \rangle|$ and the lemma holds.

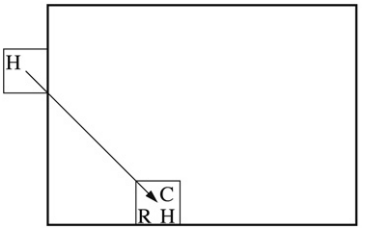

Fig. 12. Computing $H_{i+p,j+l}$ in a white block.

(iii) $i+k=i+p-l$: The HH-mp P_1 is $\langle h_{l-1}v_{l-1}\rangle$ or $\langle d^w_{l-1}\rangle$. If $P_1=\langle h_{l-1}v_{l-1}\rangle$, it becomes case (i). Now we consider $P_1=\langle d^w_{l-1}\rangle$. By definition of a path, $|\langle d^w_{l-1}d^w_1\rangle|=|\langle d^w_l\rangle|$ and the lemma holds. □

Lemma 17. *Let P_1 be an HH-mp from $(i, j+s)$ to $(i+p-1, j+l-1)$ for $0 \le s \le l-1$. Then, $H_{i,j+s}+|P_1|+|\langle d^w_1\rangle| \le C_{i+p,j+l}$ or $H_{i,j+s}+|P_1|+|\langle d^w_1\rangle| \le H_{i,j+l-p}+|\langle d^w_l\rangle|$ when $l>p$.*

Proof. Similar to the proof of Lemma 16. □

By the lemmas above, we derive that the value of $H_{i+p,j+l}$ is the maximum of the following. See Fig. 12:

(i) $R_{i+p,j+l}$
(ii) $C_{i+p,j+l}$
(iii) $H_{i+p-l,j}-l\delta$ when $l \le p$
(iv) $H_{i,j+l-p}-p\delta$ when $l>p$.

Since each value of (i), (ii), (iii) and (iv) is computed in constant time, we can compute all the values of H of the bottom row in $O(q)$ time.

3.1.4. Analysis

Given a white block with p rows and q columns, the bottom row of the block is computed in $O(p+q)$ time. The values of C of the bottom row are computed in $O(p+q)$ time and the values of R and H of the bottom row are computed in $O(q)$ time.

The rightmost column of the block is also computed in $O(p+q)$ time and thus the similarity of the white block can be computed in $O(p+q)$ time.

3.2. Black blocks

We give an algorithm for a black block. As in white blocks, we only show how to compute the values of the elements on the bottom row of the block.

3.2.1. Computing $C_{i+p,j+l}$

To compute the value of $C_{i+p,j+l}$ for $1 \le l \le q$, we need $R_{i+k,j}$ for $1 \le k \le p$, $C_{i,j+s}$ for $1 \le s \le l$, $H_{i+k,j}$ for $1 \le k \le p$, and $H_{i,j+s}$ for $0 \le s \le l$. We give two lemmas for a black block to select essential paths. Since the proofs of Lemmas 18 and 19 are similar to those of Lemmas 11 and 13, we omit them.

Lemma 18. *Let P_1 be a CC-mp from $(i, j+l-s)$ to $(i+p, j+l)$ for $1 \le s < l$. Then, there exists an element $H_{i,j+l-s}$ such that $C_{i,j+l-s}+|P_1| \le H_{i,j+l-s}+|P_2|$ where P_2 is an HC-mp from $(i, j+l-s)$ to $(i+p, j+l)$*

Lemma 19. *Let P_1 be an HC-mp from $(i, j+l-s)$ to $(i+p, j+l)$ for $0 \le s \le l$. Then, there exists an element $H_{i,j+l-t}$ for $0 \le t \le \min\{l, p-1\}$ such that $H_{i,j+l-s}+|P_1| \le H_{i,j+l-t}+|P_2|$ where P_2 is an HC-mp from $(i, j+l-t)$ to $(i+p, j+l)$.*

By Lemmas 18 and 19 and Fact 8, we can select essential paths from the top row of C and the top row of H of the block to $C_{i+p,j+l}$. The maximum paths from the leftmost column of R and the leftmost column of H to $C_{i+p,j+l}$ are all essential paths. From these, we derive that the value of $C_{i+p,j+l}$ is the maximum of the following. See Fig. 13:

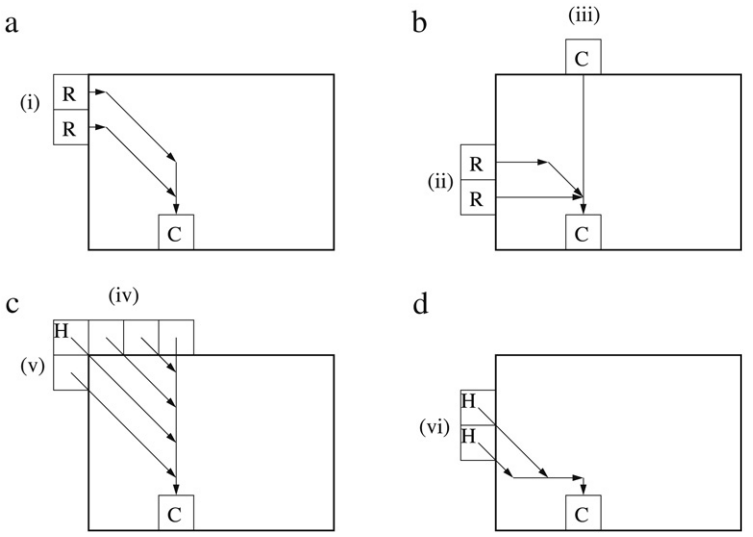

Fig. 13. Computing $C_{i+p,j+l}$ in a black block. (a) formula (i), (b) formulas (ii) and (iii), (c) formulas (iv) and (v), (d) formula (vi).

(i) $\max_{1\le s\le p-l}\{R_{i+s,j} - g_{p-s-l+1}\} - \mu + (l-1)$ for $l \le p-1$
(ii) $\max_{0\le s\le \min\{l-2,p-2\}}\{R_{i+p-1-s,j} - (l-s)\mu + s\} - g_1$ when $l \ge 2$
(iii) $C_{i,j+l} - p\mu$
(iv) $\max_{0\le s\le \min\{l,p-1\}}\{H_{i,j+l-s} + s - g_{p-s}\}$
(v) $\max_{1\le s\le p-1-l}\{H_{i+s,j} - g_{p-s-l}\} + l$ when $l < p-1$
(vi) $\max_{1\le s\le \min\{l-1,p-2\}}\{H_{i+p-1-s,j} - g_{l-s} + s\} - g_1$ when $l \ge 2$.

We need $O(p)$ time to compute the value of $C_{i+p,j+l}$.

We can compute all the values of C of the bottom row in $O(p+q)$ time. Recurrences (iii), (iv), (v) and (vi) are essentially the same as recurrences (ii), (iv), (v) and (vi) of a white block, and (i) and (ii) are similar to (v) and (vi), respectively.

3.2.2. Computing $R_{i+p,j+l}$

Computing $R_{i+p,j+l}$ for $1 \le l \le q$ in a black block is the same as in a white block. We can compute $R_{i+p,j+l}$ by recurrence (2) and it takes $O(1)$ time. Therefore, we compute all the values of R of the bottom row in $O(q)$ time.

3.2.3. Computing $H_{i+p,j+l}$

To compute the value of $H_{i+p,j+l}$ for $1 \le l \le q$, we need $C_{i+p,j+l}$, $R_{i+p,j+l}$ and $H_{i+p-1,j+l-1}$. Since we know the values of $C_{i+p,j+l}$ and $R_{i+p,j+l}$, we only need to compute the diagonal incoming value.

To compute $H_{i+p,j+l}$, we need more terms than that in a white block. Since Lemmas 14 and 15 do not hold for a black block, we need to compute paths from $R_{i+k,j}$ for $1 \le k \le p-1$ and from $C_{i,j+s}$ for $1 \le s \le l-1$.

Lemma 20. *Let P_1 be an RH-mp from $(i+k, j)$ to $(i+p-1, j+l-1)$ for $1 \le k \le p-1$. Then, $R_{i+k,j} + |P_1| + |\langle d_1^b\rangle| \le H_{i+p-l,j} + |\langle d_l^b\rangle|$ when $p \ge l$ and $R_{i+k,j} + |P_1| + |\langle d_1^b\rangle| \le H_{i,j+l-p} + |\langle d_p^b\rangle|$ when $l > p$.*

Proof. We will only prove the lemma for the case that $l > p$. For $p \ge l$, we can similarly prove the lemma.

We know that there exists an *HR*-mp P_2 from $(0,0)$ to $(i+k, j)$ such that $H_{0,0} + |P_2| = R_{i+k,j}$ and that this path passes the ith row. Let $(i, j-u)$ for $u > 0$ be the last position of the ith row which P_2 passes. The next position of $(i, j-u)$ by P_2 is $(i+1, j-u)$ or $(i+1, j-u+1)$. In other words, there exists an *HR*-mp P_3 from $(i, j-u)$ to $(i+k, j)$ such that $H_{i,j-u} + |P_3| = R_{i+k,j}$ or there exists a *CR*-mp P_4 from $(i, j-u)$ to $(i+k, j)$ such that $C_{i,j-u} + |P_4| = R_{i+k,j}$.

Suppose that there exists an *HR*-mp P_3 from $(i, j-u)$ to $(i+k, j)$. See Fig. 14(a). By Fact 8, the *RH*-mp P_1 from $(i+k, j)$ to $(i+p-1, j+l-1)$ is $\langle \hat{h}_{l-p+k} d_{p-k-1}^b\rangle$ and $R_{i+k,j} + |P_1| \le H_{i+p-1,j+l-1}$

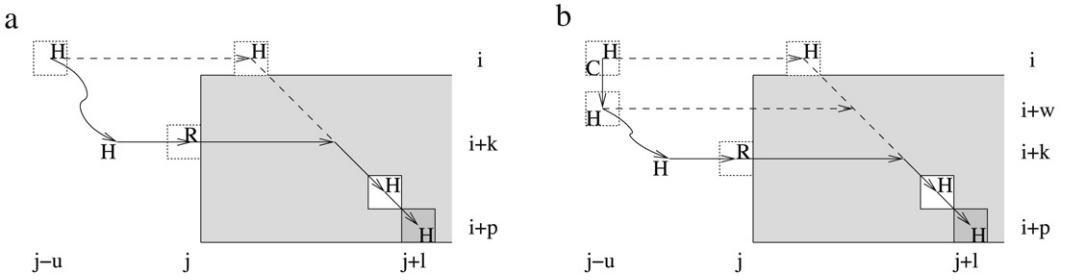

Fig. 14. Proof of Lemma 20.

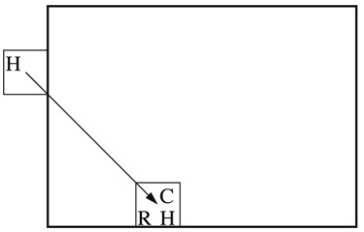

Fig. 15. Computing $H_{i+p,j+l}$ in a black block.

by Lemma 6. Thus, $H_{i,j-u} + |P_3| + |P_1| \leq H_{i+p-1,j+l-1}$. We consider the element $H_{i,j+l-p}$. By Lemma 6, $H_{i,j+l-p} \geq H_{i,j-u} + |\langle h_{l-p+u}\rangle|$ and $H_{i+p-1,j+l-1} \geq H_{i,j+l-p} + |\langle d^b_{p-1}\rangle|$. By Fact 8, an HH-mp from $(i, j-u)$ to $(i+p-1, j+l-1)$ is $\langle h_{l-p+u} d^b_{p-1}\rangle$. Therefore,

$$\begin{aligned} R_{i+k,j} + |P_1| + |\langle d^b_1\rangle| &= H_{i,j-u} + |P_3| + |\langle \hat{h}_{l-p+k} d^b_{p-k-1}\rangle| + |\langle d^b_1\rangle| \\ &\leq H_{i,j-u} + |\langle h_{l-p+u} d^b_{p-1}\rangle| + |\langle d^b_1\rangle| \\ &\leq H_{i,j+l-p} + |\langle d^b_p\rangle|. \end{aligned}$$

Suppose that there exists a CR-mp P_4 from $(i, j-u)$ to $(i+k, j)$. See Fig. 14(b). Let $(i+w, j-u)$ for $w > 0$ be the last position of the $(j-u)$th column which P_2 passes. Then, the path P_4 must pass an H-vertex at $(i+w, j-u)$. We denote P_4 by $\langle \hat{v}_w \alpha\rangle$ where α is a sequence of edges. By Fact 8, an HH-mp from $(i+w, j-u)$ to $(i+p-1, j+l-1)$ is $\langle h_{l-p+w+u} d^b_{p-w-1}\rangle$. Since $H_{i,j-u} \geq C_{i,j-u}$ by recurrence (2) and $|\langle d^b_w\rangle| > |\langle \hat{h}_w \hat{v}_w\rangle|$ by Fact 8, $H_{i,j-u} + |\langle h_{l-p+u} d^b_{p-1}\rangle| \geq C_{i,j-u} + |\langle \hat{v}_w h_{l-p+w+u} d^b_{p-w-1}\rangle|$. By Lemma 6, $H_{i,j+l-p} \geq H_{i,j-u} + |\langle h_{l-p+u}\rangle|$. Thus,

$$\begin{aligned} R_{i+k,j} + |P_1| + |\langle d^b_1\rangle| &= C_{i,j-u} + |\langle \hat{v}_w \alpha\rangle| + |P_1| + |\langle d^b_1\rangle| \\ &\leq C_{i,j-u} + |\langle \hat{v}_w h_{l-p+w+u} d^b_{p-w-1}\rangle| + |\langle d^b_1\rangle| \\ &\leq H_{i,j-u} + |\langle h_{l-p+u} d^b_{p-1}\rangle| + |\langle d^b_1\rangle| \\ &\leq H_{i,j+l-p} + |\langle d^b_p\rangle|. \end{aligned}$$

and the lemma holds for $l > p$. □

Lemma 21. *Let P_1 be a CH-mp from $(i, j+s)$ to $(i+p-1, j+l-1)$ for $1 \leq s \leq l-1$. Then, $C_{i,j+s} + |P_1| + |\langle d^b_1\rangle| \leq H_{i+p-l,j} + |\langle d^b_l\rangle|$ when $p \geq l$ and $C_{i,j+s} + |P_1| + |\langle d^b_1\rangle| \leq H_{i,j+l-p} + |\langle d^b_p\rangle|$ when $l > p$.*

Proof. Similar to the proof of Lemma 20. □

By Lemmas 20, 21, 16, 17 and Fact 8, we derive that the value of $H_{i+p,j+l}$ is the maximum one of the followings. See Fig. 15:

(i) $R_{i+p,j+l}$
(ii) $C_{i+p,j+l}$

(iii) $H_{i+p-l,j} + l$ when $p \geq l$
(iv) $H_{i,j+l-p} + p$ when $l > p$.

Since each value of (i), (ii), (iii) and (iv) is computed in constant time, we can compute all the values of H of the bottom row in $O(q)$ time.

3.2.4. Analysis

Given a black block with p rows and q columns, the bottom row of the block is computed in $O(p+q)$ time. The rightmost column of the block is also computed in $O(p+q)$ time and thus the similarity of the black block can be computed in $O(p+q)$ time.

Theorem 22. *The similarity of two run-length encoded strings in the affine gap penalty model can be computed in* $O(nm' + n'm)$ *time.*

4. Conclusion

We have presented an efficient algorithm that computes the similarity of two run-length encoded strings with affine gap penalty. To compute the similarity efficiently, we first converted the alignment problem into a path problem on a directed acyclic graph and then made new recurrences using some properties of maximum paths in this graph. Based on these recurrences and some data structures, we gave an $O(nm' + n'm)$ time algorithm for computing the similarity of two run-length encoded strings in the affine gap penalty model.

We successfully extended comparison of run-length encoded strings to a more general scoring metric. Our technique of skipping some regions using essential paths can be applicable to other problems where dynamic programming tables are divided into blocks.

References

[1] A. Apostolico, G.M. Landau, S. Skiena, Matching for run length encoded strings, Journal of Complexity 15 (1) (1999) 4–16.
[2] O. Arbell, G.M. Landau, J. Mitchell, Edit distance of run-length encoded strings, Information Processing Letters 83 (6) (2002) 307–314.
[3] H. Bunke, H. Csirik, An improved algorithm for computing the edit distance of run length coded strings, Information Processing Letters 54 (1995) 93–96.
[4] M. Crochemore, G.M. Landau, B. Schieber, M. Ziv-Ukelson, Re-use dynamic programming for sequence alignment: An algorithmic toolkit, in: C.S. Iliopoulos, T. Lecroq (Eds.), String Algorithmics, King's College London Publications, 2005, pp. 19–59.
[5] M. Crochemore, G.M. Landau, M. Ziv-Ukelson, A subquadratic sequence alignment algorithm for unrestricted scoring matrices, SIAM Journal on Computing 32 (6) (2003) 1654–1673.
[6] H. Gajewska, R.E. Tarjan, Deques with heap order, Information Processing Letters 22 (1986) 197–200.
[7] O. Gotoh, An improved algorithm for matching biological sequences, Journal of Molecular Biology 162 (1982) 705–708.
[8] D. Gusfield, Algorithms on Strings, Trees, and Sequences, Cambridge University Press, 1997.
[9] X. Huang, W. Miller, A time-efficient, linear-space local similarity algorithm, Advances in Applied Mathematics 12 (1991) 337–357.
[10] J.W. Kim, K. Park, An efficient alignment algorithm for masked sequences, Theoretical Computer Science 370 (2007) 19–33.
[11] V. Mäkinen, G. Navarro, E. Ukkonen, Approximate matching of run-length compressed strings, in: 12th Annual Symposium on Combinatorial Pattern Matching, in: Lecture Notes in Computer Science, vol. 2089, 2001, pp. 31–49.
[12] V. Mäkinen, G. Navarro, E. Ukkonen, Approximate matching of run-length compressed strings, Algorithmica 35 (2003) 347–369.
[13] J. Mitchell, A geometric shortest path problem, with application to computing a longest common subsequence in run-length encoded strings, Technical Report, Dept. of Applied Mathematics, SUNY Stony Brook, 1997.
[14] M.S. Waterman, T.F. Smith, W.A. Beyer, Some biological sequence metrics, Advances in Mathematics 20 (1976) 367–387.